Robust Edge-detection algorithm for runway-edge detection

Swathi Tandra, Zia-ur Rahman

Old Dominion University, Electrical and Computer Engineering Department
Norfolk, Virginia 23529

ABSTRACT

Fog and other poor visibility conditions hamper the visibility of runway surfaces and any obstacles present on the runway, potentially creating a situation where a pilot may not be able to safely land the aircraft. Assisting the pilot to land the aircraft safely in such conditions is an active area of research. We are investigating a method that combines non-linear image enhancement with classification of runway edges to detect objects on the runway. The image is segmented into runaway and non-runway regions, and objects that are found in the runway regions are deemed to constitute potential hazards. For runway edge classification, we make use of the long, continuous edges in the image stream. This paper describes a new method for edge-detection that is robust to the imaging conditions under which we are acquiring the imagery. This edge-detection method extracts edges using a locally adaptive threshold for the detection. The proposed algorithm is evaluated qualitatively and quantitatively on different types of images, especially acquired under poor visibility conditions. The performance of the algorithm is evaluated as a function of the signal-to-noise ratio and as a function of the visibility index. Additionally the results of our new algorithm are compared with other, more conventional edge detectors.

Keywords: I mage enhancement, Multi-scale Retinex (MSR), edge detection, edge thinning, scaling, hysteresis thresholding, line detection, blob removal, Hough transform.

1. INTRODUCTION

For past many years, different techniques have been proposed for assisting a pilot to land an aircraft safely on a wide range of airports in low visibility conditions. Examples of these technologies include, but are not limited to, millimeter wave (MMW) cameras used for producing visual-like radiometric images at real-time frame rates,[1] systems operating in the microwave region of the electromagnetic (EM) spectrum, and the infrared region of the EM spectrum.[2] Fog and low visibility conditions have hampered aviation since its inception. Low visibility conditions refer to any weather conditions such as rain, fog, and snow, other phenomenon such as haze, smoke or dust, and the lighting conditions that exist at dawn and dusk. Dense fog conditions, in particular, may present a situation that prevents the pilot from landing the aircraft safely. Fog-related accidents like the one that occurred at the Atlanta Hartsfield-Jackson International Airport on January 13th, 2005 during landing where the plane ran off the runway and became stuck in the mud [citation], along with canceled take-offs and landings have also increased. These accidents and cancellations/delays have a significant economic impact on airlines and general aviation. Pilots should be able to see the runway even when the weather conditions are conducive to deteriorating or rapidly changing. Surveillance is an important function that provides a detailed picture of all the ground traffic (aircraft and vehicles) on the runways. Typically, an airport is characterized by its runways that are of particular length and width. These runways form an elongated rectangle shape in aerial imagery that distinguishes it from its surroundings. This distinctive shape is very useful in discriminating the runway from its surroundings.

The problem of particular interest to our research is the detection of unexpected objects on a runway, especially in poor visibility. While sensor suites composed of infrared, visible and millimeter wave radar provide additional imagery to the pilot, they typically do not attempt to extract information from the imagery. This presents the pilot with yet-another-task that needs to be performed in order to fly the aircraft safely. We provide an automated technique that applies novel image processing and analysis techniques to the imagery from the sensor suite to detect the presence of some unexpected object that may present a hazard to the aircrafts during landing or take-offs. The process consists of three parts:

Contact: ST: stand003@odu.edu; ZR: zrahman@odu.edu.

1. runway edge determination to segment the runway from its surrounds;
2. detection of stationary or moving objects on the runway; and,
3. determination of whether the object on the runway represents a hazard.

In this paper, we concentrate primarily on the first task, i.e., determination of runway edges in the image frame.

Typically the sensor-pod is fixed on the nose of the aircraft, though the NASA 757 Aries aircraft had it mounted on the belly of the aircraft, just below the nose cone. When the aircraft is landing or taking-off, the flight axis is parallel to the body axis. This provides a good frame of reference for the geometrical operations that are needed to find the runway edges. It is considerably easier to discover runway edges at some airports than at others. Figure 1 shows an example approach. In this instance, the characteristics of runway are very different from the characteristics of its surrounds. The runway itself is gray with greenery around it, and two white lines mark the boundaries of the runway. Additionally, a broken white line marks the center of the runway. These two borders eventually become two intersecting lines in the image. The area above the horizon is classified as sky and the rest as ground as shown in Figure 1. So the area of interest is in-between the white boundaries and the intersection of the edge boundaries. If we can obtain these pieces of information from the imagery in an automatic manner, then we can make the task of landing an aircraft in poor visibility conditions considerably easier.

Figure 1: Runway characteristics.

Before performing any edge operation on the image it is useful to enhance it so as to reduce the impact of lighting conditions under which it was acquired. We use the non-linear, context dependent Multiscale Retinex (MSR) to perform the image enhancement (see Section 2 for a discussion of the MSR). Since the MSR does not distinguish between signal and noise, it has a tendency to amplify the noise even as it enhances the signal. It is, therefore, at times necessary to apply a noise-reduction filter such as the Gaussian low-pass filter to reduce the impact of noise amplification. This step is followed by finding the edges in the image which are then thinned to make them 1-pixel wide. Different thinning techniques are used for increasing the possibility of detection.[3–5] Thinning helps to maintain the single pixel wide edges before the final stage when thresholding is performed using hysteresis thresholding. Hysteresis thresholding has some local adaptation to image content. Additional noise reduction is performed on the edge-images to remove any unconnected edges that are smaller than a certain size. These isolated edges appear as "blobs" on the runway.[6] These unwanted elements are eliminated from the image by using masks of different sizes. Four filters are used for removing these isolated edges from the edge image, making the image more useful for automated interpretation.

Figure 2: MSR enhancement: (left) original image; (right) enhanced image.

2. APPROACH

2.1. Image Enhancement

There is a need for enhancing the image before performing any operation on it as the images acquired from UAV have varying amount of illumination depending on whether it is day or night. The image should be made ineffective of the lighting conditions. Multi-Scale Retinex (MSR) is a technique which reduces the impact of lighting conditions by making it up to a certain extent ineffective of the lighting changes which plays a major role in analyzing the image in poor visibility conditions, to improve the quality of scene image and extract better features in complex environment where the images undergo large lighting changes.[7,8] It simultaneously provides dynamic range compression, color constancy and color rendition. The enhanced image becomes independent of lighting changes under poor visibility conditions. The MSR is specified as:

$$R_i(x_1, x_2) = \sum_{k=0}^{\kappa} w_k \left(\log \left(I_i(x_1, x_2) \right) - \log \left(I_i(x_1, x_2) * F_k(x_1, x_2) \right) \right), \quad i = 1, \ldots, N, \tag{1}$$

where I_i is the i^{th} spectral band of the N-band input image, R_i is the corresponding is the Retinex output, '$*$' represents the (circular) convolution operator, F is a (Gaussian) surround function, and κ is the number of the scales. The Gaussian surround function is given by:

$$\begin{aligned} F_k(x_1, x_2) &= a_k G_k(x_1, x_2) & (2) \\ G_k(x_1, x_2) &= \exp\left(-(x_1^2 + x_2^2)/\sigma_k^2\right) & (3) \\ a_k &= \sum_{x_1, x_2} G_k(x_1, x_2), & (4) \end{aligned}$$

where the σ_k values need to be set according to the amount of enhancement required. Figure 2 (left) shows an original image of a road acquired under fairly heavy fog. The fog hampers the overall visibility of objects on the road. Figure 2 (right) shows the MSR enhanced version of the image. The MSR eliminates all of the fog-blur from the image.

2.2. Edge detection

After enhancement, the image needs to be analyzed to find any hazardous objects on the runway path. Edge detection is used for this for two reasons:

1. Edge detection has the advantage of reducing the amount of data to be analyzed, eliminating unwanted information from the image.

2. Edge detection aids in eliminating noise from the image.

In this paper we propose a new edge detection method to overcome the limitations of conventional methods. The edge detector initially performs smoothing on the image by convolving the image with the Gaussian smoothing operator. Gaussian smoothing operator is a circularly symmetric (or isotropic) mask, such that any cross-section through its center yields a weight profile that has the form of a Gaussian or normal curve. Its kernel is in the form of a Gaussian hump. It removes small-scale texture and noise as effectively as possible for a given spatial extent in the image. Next, the edge magnitude of a pixel is computed by taking the difference between the mean intensities of two groups of pixels in the regional difference matrix, D, given by:

$$D = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} = \begin{bmatrix} I_{x_1-1,x_2-1} - I_{x_1,x_2} & I_{x_1-1,x_2} - I_{x_1,x_2} & I_{x_1-1,x_2+1} - I_{x_1,x_2} \\ I_{x_1,x_2-1} - I_{x_1,x_2} & I_{x_1,x_2} - I_{x_1,x_2} & I_{x_1,x_2+1} - I_{x_1,x_2} \\ I_{x_1+1,x_2-1} - I_{x_1,x_2} & I_{x_1+1,x_2} - I_{x_1,x_2} & I_{x_1+1,x_2+1} - I_{x_1,x_2} \end{bmatrix} \tag{5}$$

All $d_{ij}, i, j = 1, \ldots, 3$ with intensity values above a (pre-defined) threshold belong to the first group, G_a, and all others belong to the second, G_b. The center pixel in the 3×3 neighborhood , I_{x_1,x_2}, is assigned a value equal to the difference of the means of the two groups:

$$M_{x_1,x_2} = M(x_1, x_2) = \frac{1}{N_a} \sum_{\substack{k_1=1 \\ d_{k_1,k_2} \geq \tau}}^{3} \sum_{k_2=1}^{3} d_{k_1,k_2} - \frac{1}{N_b} \sum_{\substack{k_1=1 \\ d_{k_1,k_2} < \tau}}^{3} \sum_{k_2=1}^{3} d_{k_1,k_2} \tag{6}$$

where N_a, N_b are the number of elements in G_a and G_b respectively, and $M(x_1, x_2)$ represents the edge magnitude of a pixel at (x_1, x_2). Figure 3 (left) shows an example of applying this method to a test image.

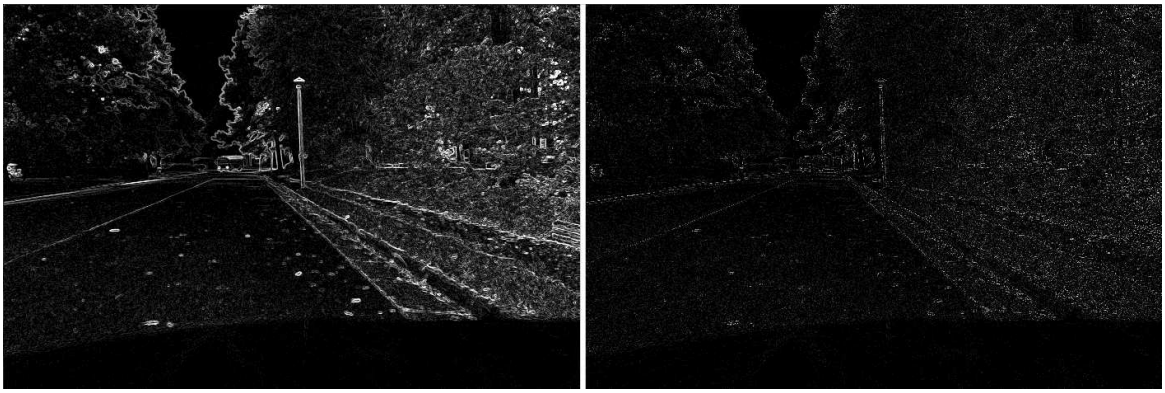

Figure 3: (left) Gradient image; (right) Edge thinned image.

2.3. Edge thinning

The gradient image produced by the method described in Section 2.2 produces edges that are more than one pixel thick. These edges need to be reduced in thickness for computational efficiency. We perform an edge-thinning operation that reduces edges to unit thickness while maintaining the original connectivity of edges. Edge thinning is performed by computing the gradient magnitude in four directions—horizontal (x_1), vertical (x_2), 45° and 135°. The four gradients are computed using:

$$I_{x_1} = \sqrt{\left(I(x_1, x_2) - I(x_1, x_2 - 1)\right)^2 + \left(I(x_1, x_2) - I(x_1, x_2 + 1)\right)^2} \tag{7}$$

$$I_{x_2} = \sqrt{\left(I(x_1, x_2) - I(x_1 - 1, x_2)\right)^2 + \left(I(x_1, x_2) - I(x_1 + 1, x_2)\right)^2} \tag{8}$$

$$I_{45^\circ} = \sqrt{\left(I(x_1,x_2) - I(x_1-1,x_2+1)\right)^2 + \left(I(x_1,x_2) - I(x_1+1,x_2-1)\right)^2} \quad (9)$$

$$I_{135^\circ} = \sqrt{\left(I(x_1,x_2) - I(x_1-1,x_2-1)\right)^2 + \left(I(x_1,x_2) - I(x_1+1,x_2+1)\right)^2} \quad (10)$$

Edges in the edge-image are made one pixel thick by considering a 3×3 neighboring region around an edge pixel. If any of the following conditions are met, then the pixel is considered to be an edge-pixel, otherwise it is set to zero:

$$\begin{aligned}
I(x_1,x_2) &\geq \max_{(k_1,k_2)} I(x_1+k_1,x_2+k_2), \quad k_1,k_2 \in \{-1,0,+1\}, \\
I(x_1,x_2) &\geq \max_{(k_1,k_2)} I(x_1+k_1,x_2+k_2), \quad k_1 = 0, k_2 \in \{-1,0,+1\} \wedge I_{x_1}(x_1,x_2) \geq I_{x_2}(x_1,x_2) \\
I(x_1,x_2) &\geq \max_{(k_1,k_2)} I(x_1+k_1,x_2+k_2), \quad k_1 \in \{-1,0,+1\}, k_2 = 0 \wedge I_{x_2}(x_1,x_2) \geq I_{x_1}(x_1,x_2) \\
I(x_1,x_2) &\geq \max_{(k_1,k_2)} I(x_1+k_1,x_2+k_2), \quad (k_1,k_2) \in \{(-1,-1),(0,0),(+1,+1)\} \wedge I_{45^\circ}(x_1,x_2) \geq I_{135^\circ}(x_1,x_2) \\
I(x_1,x_2) &\geq \max_{(k_1,k_2)} I(x_1+k_1,x_2+k_2), \quad (k_1,k_2) \in \{(+1,+1),(0,0),(-1,-1)\} \wedge I_{135^\circ}(x_1,x_2) \geq I_{45^\circ}(x_1,x_2).
\end{aligned} \quad (11)$$

Then after performing thinning scaling is performed on the image since the image intensities produced after thinning are so small that they need to be normalized before performing any thresholding operation on them. Figure 3 (right) shows the result of performing thinning on the image gradient. The resultant image typically has very low intensity values which need to be normalized for spreading the pixel intensities over the entire range of intensities. An "autolevels" process is used to scale the intensity values to cover the full 8-bit display gamut.

2.4. Thresholding

Thresholding is performed on the scaled image using Hysteresis thresholding which results in a binary image. Two threshold values are specified: an upper threshold, τ_u, and a lower threshold, τ_l.

$$I(x_1,x_2) = \begin{cases} 255, & I(x_1,x_2) > \tau_u \\ 0, & I(x_1,x_2) < \tau_l \\ 255, & \tau_l < I(x_1,x_2) < \tau_u \wedge \max\limits_{(k_1,k_2)\neq(0,0)} I(x_1+k_1,x_2+k_2) > \tau_u, k_1,k_2 \in \{-1,0,+1\} \end{cases} \quad (12)$$

The upper threshold, τ_u is used to find strong edges and the lower threshold, τ_l is used to find weak edges. Hysteresis thresholding simultaneously attempts to eliminate weak edge points that are caused by noise while filling in gaps between strong edge points where only weak edge response is detected. In order to automate the whole process, thresholding is done by setting $\tau_u = 25$ and $\tau_l = 12$— approximately half of τ_u. The threshold values are always changed using a ratio of 2:1 which is considered standard for this algorithm. Various ratios and values were tried and an optimum value was found which suited all kind of images. Figure 4 (right) shows the effect of thresholding the scaled image in the ratio (2:1), with $\tau_u = 25$ and $\tau_l = 12$.

2.5. Removal of Isolated edges

After edge-detection, thinning, and hysteresis thresholding, there are still isolated edges in the image that can be removed by applying a series of blob-removal filters. These edges should be removed from the image because they do not provide any useful information in detection runway edges or hazards on the runway. Extracting edges is essential for identifying or interpreting meaningful physical objects from images when analyzing an image, an appropriate balance between too many and too few edges is difficult to find. Under-detection which is detection of too few edges will lead to inadequate information without any strong evidence about the edges whereas over-detection will give too many edges which cause confusion and extra hypotheses and extra computation. So these extra data values which can be classified as line-blobs. These blobs are points or regions in the image that are either brighter or darker than the surrounding.

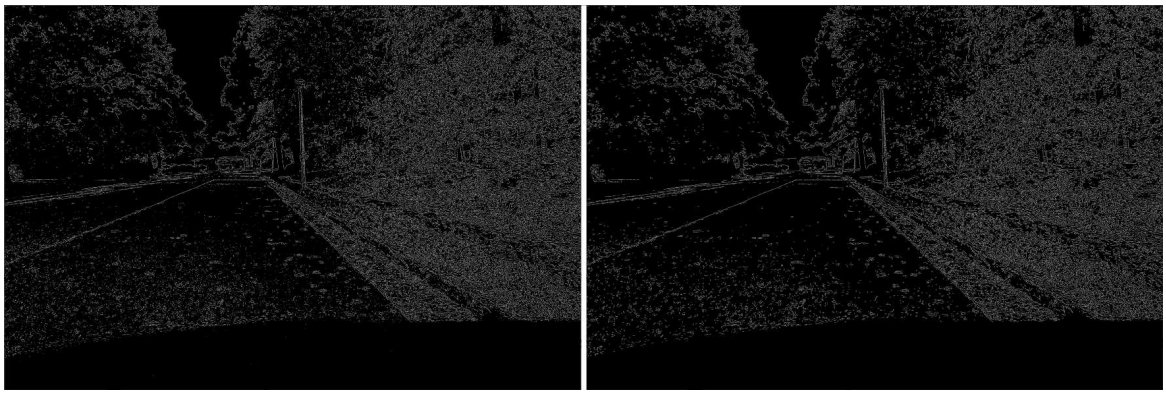

Figure 4. (left) Hysteresis thresholding is applied to the scaled image with Upper and Lower thresholds values 25 and 12; (right) After removal of isolated edges from the image.

The first filter that we use for blob-removal considers a 7×7 region in the edge-image. We assume that the blob is contained in the central 3 region within this 7×7 region. So an easy test to remove blobs is to check the number of zeros, $\mathcal{Z}_{7,3}$, in the 7×7 region excluding the central 3×3 region. If $\mathcal{Z}_{7,3} > \tau_{7,3}$ then all the values within the 3×3 central region are set to zero. Here, $\tau_{7,3}$ is a predefined threshold. Typical values of $\tau_{7,3}$ range between 38–40. This process does not affect the strong (continuous) edges but eliminates regions that have isolated edge pixels. Figure 5 (left) shows the 7×7 region: the central blob profile is marked by the symbol X.

After the application of the first filter, there are still isolated edge-pixels that have not been completely eliminated. A second filter of size 11×11 is applied to the output of the first filter. This filter eliminates slightly larger isolated edges or blobs. A similar process as that described for the first filter is applied, except the comparison in this case is: $\mathcal{Z}_{11,5} > \tau_{11,5}$, where $\mathcal{Z}_{11,5}$ is the number of zeros in the 11×11 region excluding the central 5×5 region, and $\tau_{11,5}$ a predetermined threshold value. When this comparison is valid, all the pixels inside the 5×5 region are set to zero. Typical values of $\tau_{11,5}$ range between 90–96. Figure 5 (right) shows the 11×11 mask.

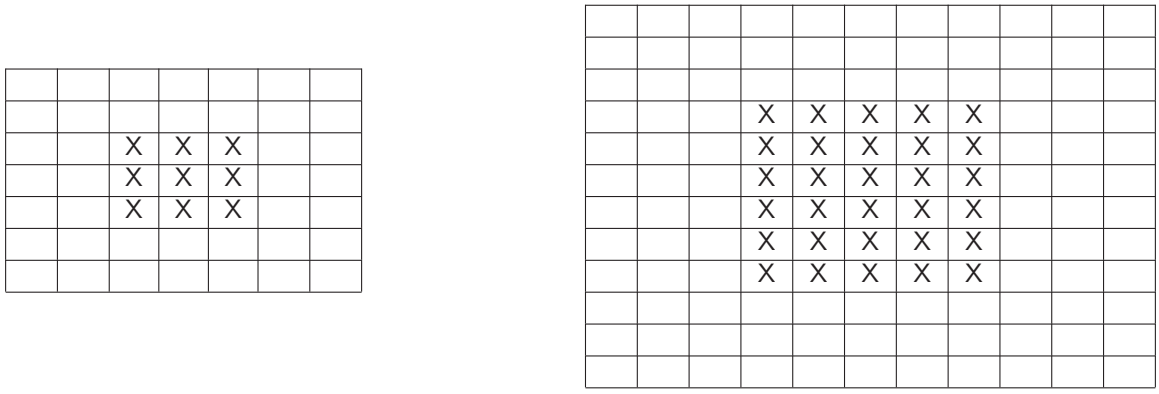

Figure 5: (left) 7×7 region with a 3×3 blob region; (right) 11×11 region with a 5×5 blob region.

In addition to the square regions, there are also isolated edge-artifacts that have one dimension that is larger than the regions considered in the first two filters. For removing these vertical and horizontal line-blobs, we apply two additional filters. The first filter considers a 13×7 region and excludes the center 7×3 pixels count in determining the number of zero pixels, $\mathcal{Z}_{13,7}$, surrounding the edge. If $\mathcal{Z}_{13,7} > \tau_{13,7}$ then all the pixels in

the 7×3 are set to zero. Figure 6 (left) shows the 13×7 mask used for removing long vertical line-blobs. For removing horizontal line-blobs a 7×13 mask is considered and the center 3×7 region is excluded. If the number of zeros, $\mathcal{Z}_{7,13} > \tau_{7,13}$, then all the pixels in the 3×7 region are set to zero. Figure 6 (right) shows the 7×13 mask used for removing long horizontal line-blobs.

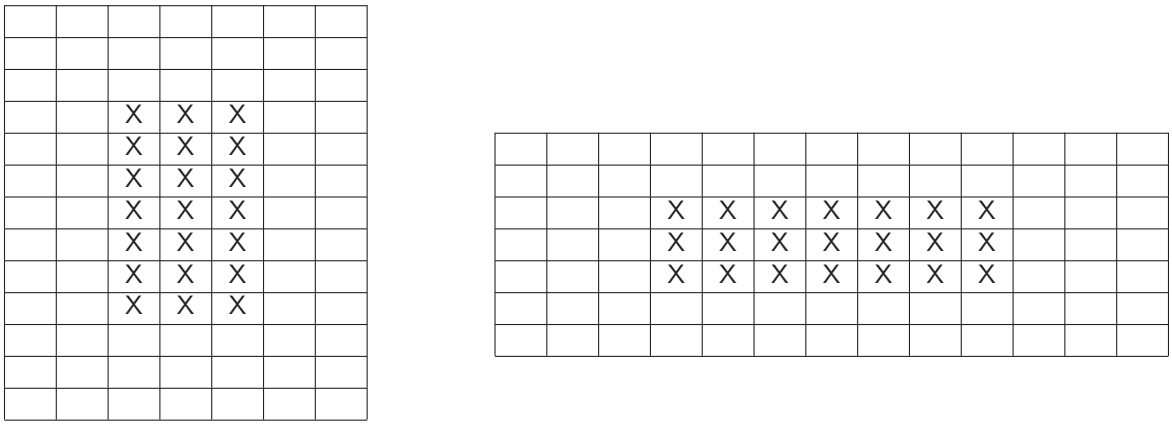

Figure 6: (left) 13×7 region with a 7×3 blob region; (right) 7×13 region with a 3×7 blob region.

Figure 7 shows a close-up of a region from the image shown in Figure 4 (left) and Figure 8 shows the result of applying the blob-removal filters. As can be seen in Figure 8, most of the isolated edges are removed by using these four blob-removal filters in cascade. We will assume that the objects that pose a hazard to safe landing of an aircraft are have a projection that is greater than 13×13 pixels. These objects are left unaffected from this filtering method and can easily be detected and judged by the pilot if they are hazardous or not to the aircraft.

2.6. Outlining runway edges

The output edge-image consists of long continuous edges along with other information that is not useful for detecting the runway edges. We assume that the runway edges are the longest edges in the edge-image and that edges smaller than a particular length are not of interest in finding hazards on the runway. The clear candidate for detection continuous edges that have a minimum length is the Hough transform which has been shown to eliminate edges that are smaller than a particular size.[9] Hough transform[10, 11] is very useful for detecting these long runway edges because it is not too sensitive to imperfect data or noise. Each pixel point in the image corresponds to a sinusoidal curve in the parameter space. Points lying on the same line form sinusoidal curves

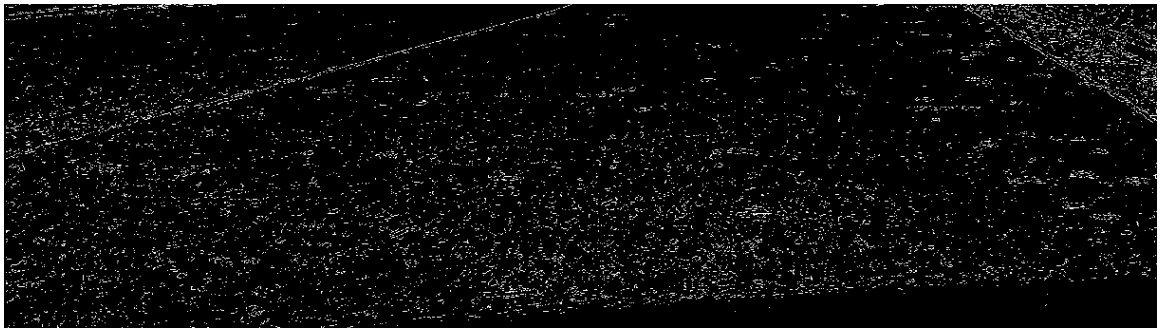

Figure 7: Part of the thresholded image is considered to view the blobs occurring on the path.

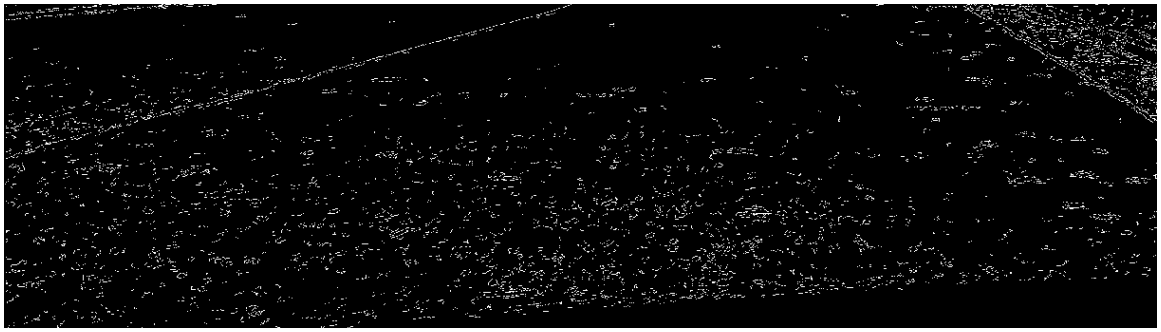

Figure 8: The blobs are eliminated to an extent by using the filtering process.

in the parameter space intersecting at the same point. The coordinates of the point of intersection define the parameters of the line in the Cartesian space. A straight line is represented in the polar form by the equation 13.

$$r = x_1 \cos(\theta) + x_2 \sin(\theta) \tag{13}$$

A line is transformed to a point using this formula. Hough space makes use of an accumulator array $\mathcal{C}$. The angle parameter θ takes values in the range $\pm 90^\circ$ and ρ takes the values in the range 0–$\sqrt{MN}$, where M is the number of rows, and N is the number of columns in the image. Each pixel of the image is mapped into a set of points in the accumulator array $\mathcal{C}$. Each cell in the accumulator array corresponds to a straight line with a particular orientation and intercept. The number of points in each cell of is an indication of the length of the corresponding straight edge. The accumulator array thus provides a histogram in the polar-space of the number of points that belong to lines of different orientations and intercepts. This allows us to eliminate all those lines that have fewer than a minimum number of points. An inverse mapping from the Hough to the image domain then allows us to draw the "significant" edges in the image. The results of implementing Hough transform as shown in the Figure 10. The image shows the significant edges of Figure 9 highlighted in bold. As stated earlier, the long continuous edges retrieved this allow us to demarcate the boundaries of the runway clearly, making the task of detecting hazards on the runway easier.

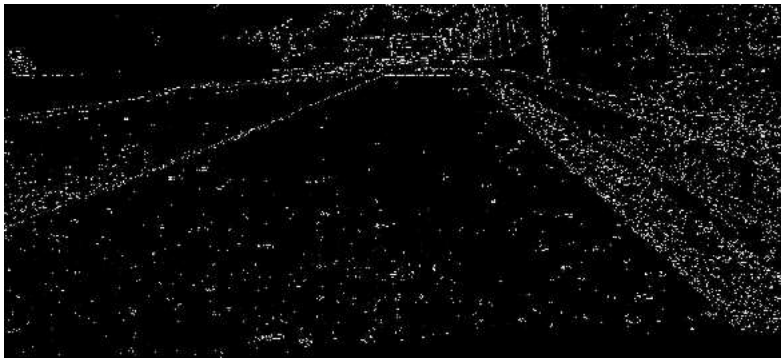

Figure 9: Part of the image where the road can be seen clearly.

3. COMPARISON OF DIFFERENT EDGE DETECTORS

Performance of the new edge detector is evaluated by comparing it with several conventional operators like Roberts, Prewitt, Sobel[12] and Canny.[13] Sobel, Roberts and Prewitt are matrix-area gradient operations that determine the level of variance between the center and its neighboring pixels. The edge-detection operator is

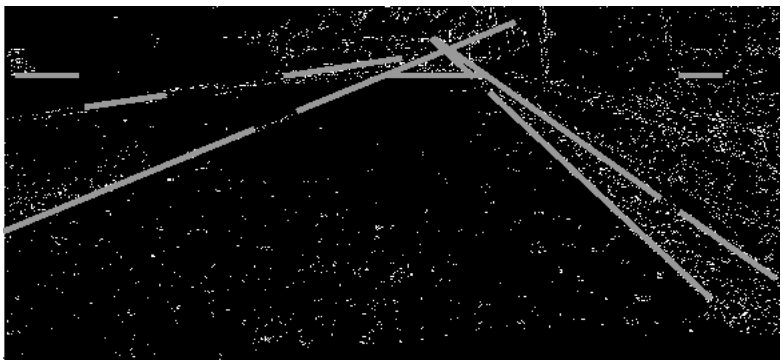

Figure 10: Result of applying Hough transform.

calculated by forming a matrix centered on a pixel chosen as the center of the matrix area. If the value of this matrix, area is above a given threshold then the middle pixel is classified as an edge otherwise as a non-edge. The Canny operator finds edges by minimizing the error rate, marking edges as closely as possible to the actual edges to maximize localization and marking edges only once when a single edge exists for minimal response. The response of the operators and the new edge detector are shown in Figure 11.

4. CONCLUSIONS

In this paper we have presented a new edge detection method which shows much promise in the field of runway detection. This technique helps in better vision of the runway path as it removes unwanted spurious noise by filtering. The amount of information required by the pilot for processing has also been decreased. This method is compared with other edge detectors and the performance of it is better compared to other operators in terms of computational time and the number of edges wanted in this particular application. Computational time plays a very major role in this application as timely view of the hazardous objects helps in stopping from any damage to the aircraft. Moreover the threshold is set to an automated value which is a must as each time the user need not have to enter it. This makes the whole system as automated. The output image is classified as runway region and non-runway region which helps in detecting any objects present on the runway path.

ACKNOWLEDGMENTS

The authors wish to thank the NASA Aviation Safety Program for the funding which made this work possible. This work was supported under NASA cooperative agreement NNL07AA02A.

REFERENCES

1. M. Shoucri, R. Davidheiser, B. Hauss, P. Lee, M. Mussetto, S. Young, and L. Yujiri, "A passive millimeter wave camera for aircraft landing in low visibility conditions," *IEEE Aerospace and Electronic Systems Magazine* **10**, pp. 37–42, May 1995.
2. V. Norris, "System for enhancing navigation and surveillance in low visibility conditions," *United States Patent 5,719,567* , 1998.
3. J. Park, H. C. Chen, and S. T. Huang, "A new gray level edge thinning method," in *Proceedings of the ISCA 13th International Conference on Computer Applications in Industry and Engineering*, pp. 114–119, November 2000.
4. A. Martin and S. Tosunoglu, "Image processing techniques for machine vision," in *Conference on Recent Advances in Robotics*, pp. 1–9, 2000.
5. U. Y. Desai, M. M. Mizuki, I. Masaki, and B. K. P. Horn, "Edge and mean based image compression," *MIT AI Memo #1584* , 1996.

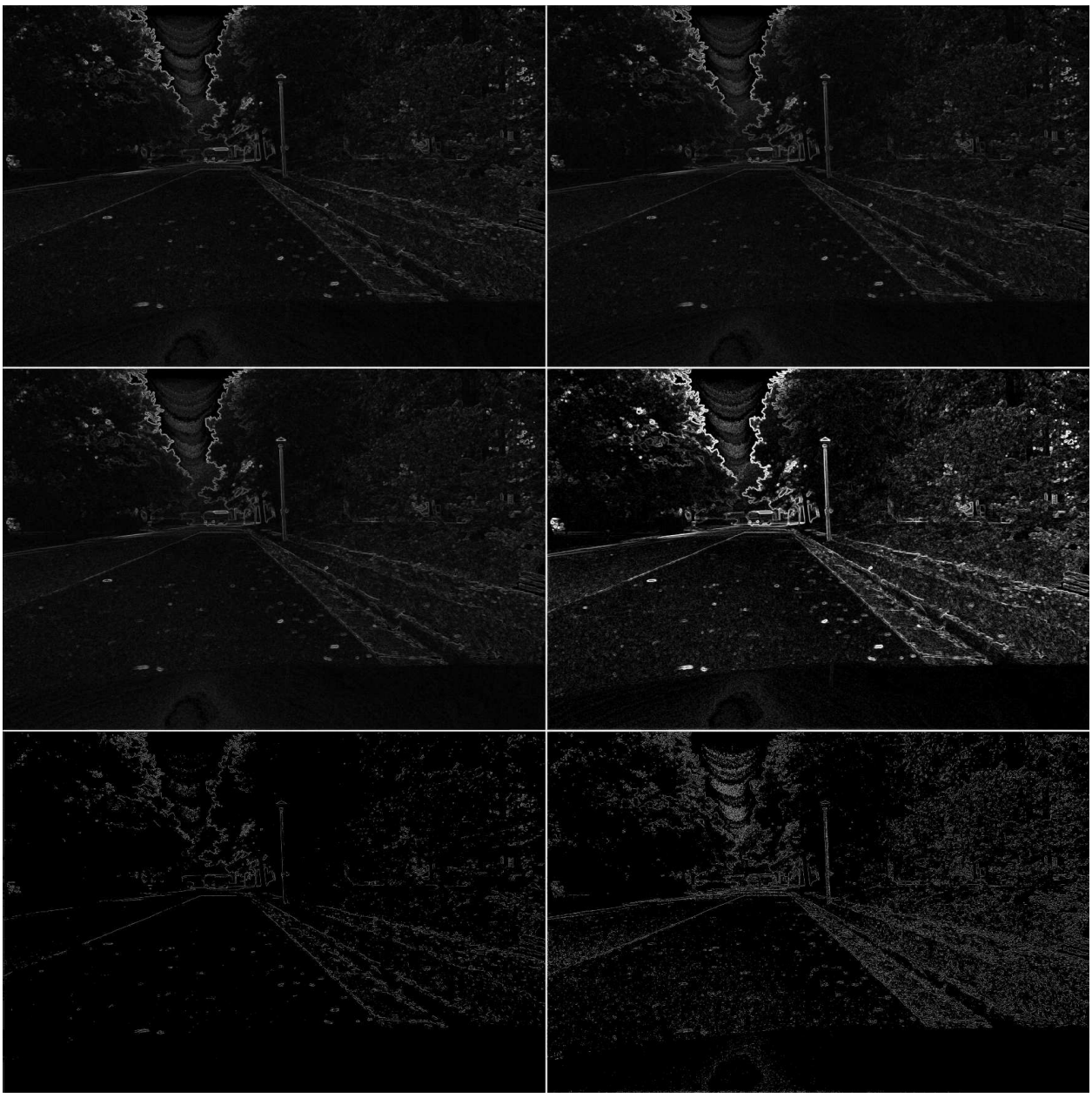

Figure 11. (top-left) Sobel; (top-right) Prewitt; (middle-left) Roberts; (middle-right) New edge detector; (bottom-left) Canny with $\tau_u = 8; \tau_l = 4$; (bottom-right) New edge detector $\tau_u = 25; \tau_l = 12$.

6. X. Zheng and Q. Gao, "Image noise removal using perceptual edge features," *ICGST International Journal on Graphics, Vision and Image Processing* , pp. 1–8, March 2006. Special Issue on Edge Detection and Tracking.
7. Z. Rahman, D. J. Jobson, and G. A. Woodell, "Retinex processing for automatic image enhancement," *Journal of Electronic Imaging* **13**(1), pp. 100–110, 2004.
8. Z. Rahman, D. J. Jobson, G. A. Woodell, and G. D. Hines, "Automated, on-board terrain analysis for

precision landings," in *Visual Information Processing XV*, Z. Rahman, S. E. Reichenbach, and M. A. Neifeld, eds., Proc. SPIE 6246, 2006.
9. Z. Rahman and D. J. Jobson, "Noise, edge extraction and visibility of features," in *Visual Information Processing XIV*, Z. Rahman, R. A. Schowengerdt, and S. E. Reichenbach, eds., Proc. SPIE 5817, 2005.
10. P. V. C. Hough, "Method and means for recognizig complex patterns," *U.S. Patent 3,069,654* , 1962.
11. D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition* **13**(2), pp. 111–122, 1981.
12. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, Reading, MA, 1993.
13. J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6), pp. 679–698, 1986.